ECE 657A Assignment #2

Yuzhou Wang (20609396), Laura McCrackin (20262085),and Huang Tianhui (20587328)

March 10, 2016

1 Parameter Selection and Classification (for dataset D)

1.1 question 1

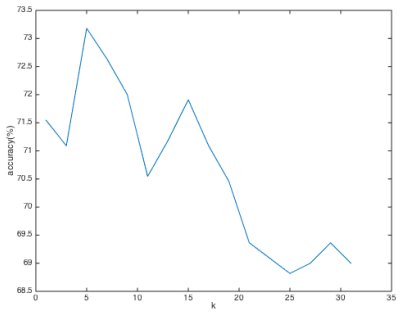
For the original dataset D, we notice that there is no missing values, the value is around the average mean, so we could use Z-score normalization to preprocessing data and use the function zscore in Matlab library directly. Our group use the first half of data for training, the second for testing.

1.2 question 2

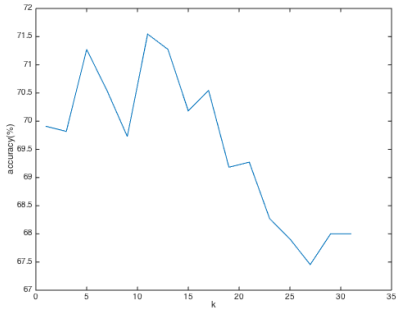
For 5-fold cross validation, we could use the function "crossvalind", and use "fitcknn" to find K-NN neighbours. And for the calculation of accuracy, we could use "predict" function to judege the reality result is how similar with the therotical result. And insetead of the loop, everytimes we compare the best k, so that we could get the best K. For run time calculate, we could use "tic", "toc" function directly. By spliting data into training data and test data, and judge the result, we could get the truth positive, truth negarive, false positive and false negetive respectively. For more details, could refernce the code.

Here we could see these two pictures for K-NN, and some different result could get. For the above four pictures, we could notice that everytime the best k changes, the value of best k is around 5-12 and the accuracy is between 71% and 73%.

We could also notice that when k is smaller than the range of best k, the accuracy increased and later larger the largest best k, it decreased. I think maybe it is the reason that less than k points is to small to be a class, in other words, less points can not represent a class, but if K is too large, then it means that some other classes's points maybe be included, which leads to a lower accuracy. Lastly, since we randomly choose the dataset and split them so that there could be some flexible results also. And what is more, we should also notice about the accuracy about the dataset itself, maybe there are some outliers. And what is more, since it is DNA sequence, it means some cuts may represent nothing, but it still appears at the dataset with a label. Above all, the result we get maybe not the exactly right answer.



(a) best k=5



(b) best k=12

1.2.1 question 3

Since the purpose of SVM is to produce a classifier that will work well on unseen examples (generalizes well), so it belongs to the decision (function) boundary approach. For design details, we could use two "for" loops, we could generate different combinations of "c" and "gamma". Firstly, we use the "sprintf" function to deal with different c and gamma as input. Later use "libsvmtrain" to training the data, use "libsvmpredict" to predict the data, and for calculating the trueth positive part, similar with the above.

According to the best c and gamma, we could trace the ROC plot.C is the cost of classification as correctly stated, a large C gives you low bias and high variance. Low bias because you penalize the cost of missclasification a lot. A small C gives you higher bias and lower variance. Gamma is the parameter of a Gaussian Kernel (to handle non-linear classification). For non-linear classification, gamma controls the shape of the "peaks" where you raise the points. A small gamma gives you a pointed bump in the higher dimensions, a large gamma gives you a softer, broader bump.So a small gamma will give you low bias and high variance while a large gamma will give you higher bias and low variance. Theoretically with the increasing of gamma(kennel width), the accuracy decrease, because if the gamma is small enough(nearly 0), then it means that data point is not influenced or correlated with other data points. Small kernel width may cause over-fitting, and large one under-fitting. The so-called optimal kernel width is merely selected based on the tradeoff between under-fitting loss and over-fitting loss.

1.2.2 question 4

Since we need to calculate the data result for 20 times, so define different matrix result and all the initial values are all 0. For Naive Bayes classifier function could use "fitNaiveBayes" function directly, the training process or calculate true positive are all similar to the above method. And for decision tree classifier could use function "fitctree" and method also similar to the above. For neural network, we could first create a neutal network by the function of patternnet and hidden layer by default is 10. For training method, we could use "train" function, and label different networks according to the test result, by setting and labelling class by hand, we could better compare with test set.

From the accuracy table above, we could see that the best method is decision tree with the highest accuracy, precision,recall and F-measure but it is also the waste most of time to decide. The worst answer result is from K-NN method.

For neural network, we could see the result in picture "perfomance". Generally, the error reduces after more epochs of training, but might start to increase on the validation data set as the network starts overfitting the training data. In the default setup, the training stops after six consecutive increases in validation error, and the best performance is taken from the epoch with the lowest validation error. Next we could see figre "neural-confusion". Since result outputs = targets. The R value is an indication of the relationship between the outputs and

targets. If R = 1, this indicates that there is an exact linear relationship between outputs and targets. If R is close to zero, then there is no linear relationship between outputs and targets. We could see the result for test, calidation and train group, the result is around 50%, so it is not really good or not really bad. For ROC graph, we could see the figure "neural-receiver", for training, validation and test ROC, they all have better perfomance since it is above 0.5. For neural-gradient figure, we could see that the trend of grandient is nearly stable, since it represents the slope of the tangent of the graph of the function. It means that there is no significant increasing or decreasing(this picture not sure). From the figure "netural-error", we could see that around the 0 errors, it has the best result.(this picture also not sure) For this example, the training data indicates a good fit. The validation and test results also show R values that greater than 0.9.

	K-NN	SVM	Naive Bayes	Decesion tree	Neural Network
Accuracy	74.4182	89.1864	86.4864	92.9818	82.5773
std accuracy	1.2322	0.7382	0.6734	0.9843	1.7322
Precision	0.9104	0.9113	0.8603	0.9391	0.8461
std Precision	0.028	0.0131	0.0101	0.0146	0.0240
recall	0.5415	0.8764	0.8810	0.9250	0.8089
std recall	0.0293	0.0097	0.0104	0.0126	0.0330
F-measure	0.6782	0.8934	0.8705	0.9319	0.8265
std F-measure	0.0195	0.0070	0.0060	0.0093	0.0186
Training time (S/Sample)	0.0081	0.1028	0.0093	0.00518	0.2698
Classification Time(Sample)	0.008	0.0690	0.0026	0.0104	0.0082

1.2.3 question 5

For test accuracy, it could be reflected by accuracy, precision and f-measure. At the same time we could also combine the std to see the stable result.Precision is how useful the search results are, and recall is how complete the results are.high precision means that an algorithm returned substantially more relevant results than irrelevant, while high recall means that an algorithm returned most of the relevant results.Recall is defined as the number of relevant documents retrieved by a search divided by the total number of existing relevant documents, while precision is defined as the number of relevant documents retrieved by a search divided by the total number of documents retrieved by that search. For f-meansure, it considers both the precision p and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results, and r is the number of correct positive results divided by the number of positive results that should have been returned. The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst at 0.

From the above table, we could get the conclusion that Decision tree method could get the highest accuracy. The std accuracy is also has a relatively low level, so it means the average result is relatively stable. For the precision and std precision, all of the methods have a good performance, but maybe the best one is still the decision tree method.For recall and std recall, we could also

get the conclusion that decision tree method and neural network have better perfomance, which means that the thue positive divided by the correct answer for both true positive and false negative is high. Similarly, the decision tree method also have the good performance for f-measure and std-measure.

For classification time in K-NN, it should compare all test samples with test sampeles. For time percepective, naive bayes has the best perfomance and also have the lowest time value.

So we could choose decision tree method for both time saving and higher accuracy.

2 Clustering Analysis (for dataset F)

2.1 question 1

1) Perform hierarchical clustering using agglomerative algorithms:
a)
For clustering data into 10 classes, we could use the matlab library function 'clusterdata'. Firstly, for separation-index, we should calculate the center of each cluster and calculate the square of each distance for each point with the center point. Later, we could use the result to divided by the total number of rows muliply the maximum distance for each cluster. And for rand-index, it should be calculated by (a+d)/M, here 'a' means TP and 'd' means FN, M means total number of comparation. For f-measure, we should firstly caluculate the number of objects in class j and cluster i, at the same time should also calulate the number of objects in class i as ni and number of objects in class j as nj. Later we could get the result of precision and recall. Lastly we could get the result using the function: F(i,j) = 2*precision(i,j)* recall(i,j)/(precision(i,j)+recall(i,j)). For different algorithm, should use the function of 'single', 'complex' and 'ward'.

And according to the definition, we could get the conclusion that the result of separation-index is highly depending on the cluster result. Smaller separation-index means that every point in the cluster is close and as a result there could be more groups. The result of rand-index and f-measure could have the similar trend for representing accuracy of the same cluster group, which is corresponding different with separation-index. It means that if there is a higher accuracy for cluster, the rand-index and f-measure could correspondingly higher, but it means larger size of cluster, as a result separation-index could be lower.

For separation-index there are two ways to calculte, one is for the maxium distance and another is for minimum distance. Below the first table shows the result for the maxmium distance. For maximum distance method:

	single	complete	ward
Separation-Index	1.7540	0.6151	0.4937
Rand-Index	0.1097	0.8184	0.8772
F-measure	0.0962	0.2168	0.2801

For separation-index, according to the function, we could get the conclution that when we choose the maximum method, the samller the value it is, the

closer for a class and far away for different class. So we hope the SI value is samll.The trend should be different with rand-index and f-measure.So, the ward algorithm have the best result for the samllest separation-index and the largest rand-index and f-measure value. b)

After we run the 'ward' algorithm, and could the result for best cluster number is 15.

2.2 question 2

a)

For calculation method of separation-index, rand-index and f-measure, they are similar with the above. The algorithm of k-means could be run by the function kmeans, and we could see the picture for 2-2-(a). We could see that both rand-index increase a lot with the increasing number of cluster, and f-measure increase a lot at the beginning and later a little bit decrease, but separation-index decrease.

The reason why rand-index increase is because of the calculating function with the increasing number of cluster, it is closer to the right cluster way, but for the value of M it did not change, so it has an increasing trend. Similarly, f-measure also increase, for increasing number of clusters, it could reflect better accuracy. And for the separation-index, with the increasing number of cluster the value of SI should also decrease for more densy in the same group and more distance for different class. Above all, all the three methods shows that the increasing number of clusters leading to a better result.

b)

Combined with three methods, the best cluster number is 8, without too high separation-index or too low rand-index and f-measure.

2.3 question 3

a)

For the fuzzy c-means methods, we could use 'fcm' function directly, and label the result is same with digit '1' or '3', so that we could get the graph below.

From the graph, we could see that, for digit 1, it mainly in cluster 5 to 7, without too much overlapping with other groups. However, the result of digit 3 could be checked in different clusters, which means that there are more overlaps in cluster 3 but less in cluster 1.For the reason why there is less overlapping for digit 1 maybe for the hand writting dataset, the digit '1' is hard to make confusing with other digits unlike digit '3',

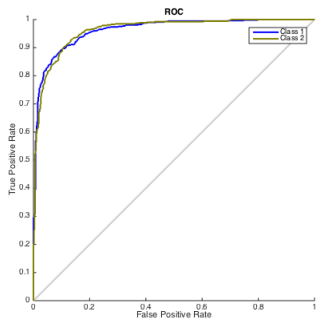
b)

In fuzzy clustering (also referred to as soft clustering), data elements can belong to more than one cluster, and associated with each element is a set of membership levels.But in hard clustering, data is divided into distinct clusters, where each data element belongs to exactly one cluster.

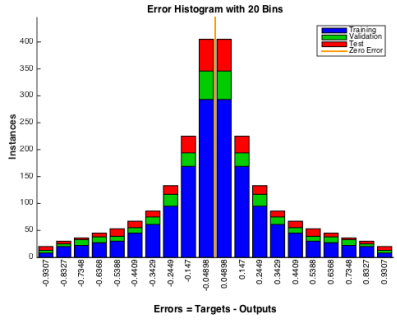
For Fuzzy c-means
Separation-index=0.4454
Rand-Index=0.8905
F-measure=0.2844

For linkage method=single
Separation-Index=1.7540
Rand-Index=0.1097
F-measure= 0.0962
For linkage method=complete
Separation-Index=0.6151
Rand-Index=0.8184
F-measure=0.2168
For linkage method=ward
Separation-Index=0.4937
Rand-Index=0.8772
F-measure= 0.2801

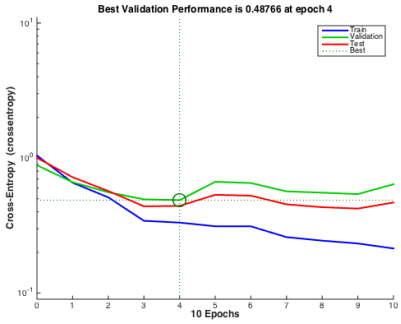
So from the above, we could get the conclution that, fuzzy-c means has a better perfomance than other three methods.


ROC
Class 1
Class 2
True Positive Rate
False Positive Rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.2
0.4
0.6
0.8

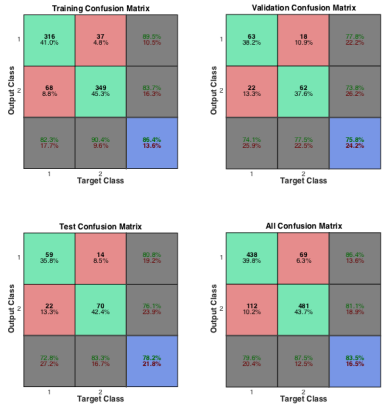
(a) ROC



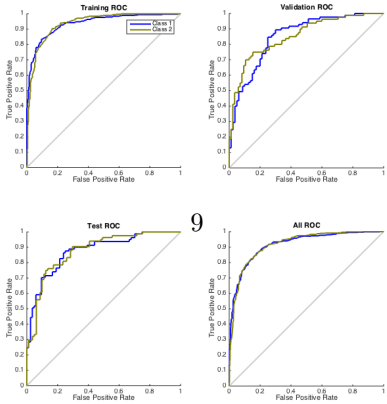
(a) netural-error



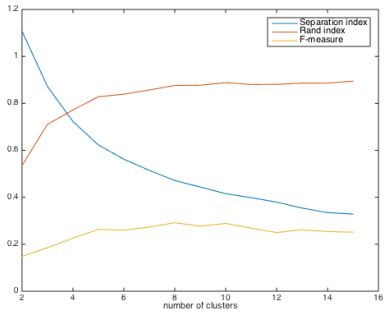
(b) perfomance



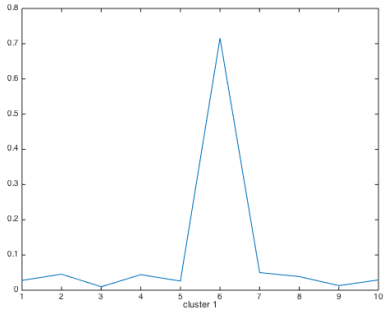
(c) neural-confusion



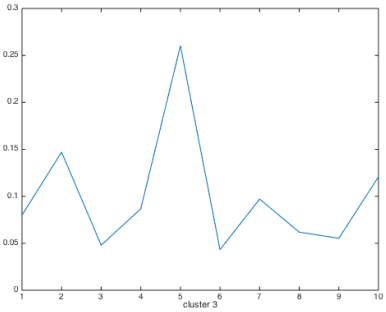
(d) neural-receiver



(a) k-means algorithm



(a) digit 1



(b) digit 3